AFRL-IF-RS-TR-2003-199
Final Technical Report
December 2003

COMMAND POST OF THE FUTURE

CYCORP, Incorporated

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J487

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-199 has been reviewed and is approved for publication

APPROVED: /s/

RAYMOND A. LIUZZI
Project Engineer

FOR THE DIRECTOR: /s/

JAMES A. COLLINS, Acting Chief
Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
DECEMBER 2003

3. REPORT TYPE AND DATES COVERED
Final Feb 99 – Feb 03

4. TITLE AND SUBTITLE
COMMAND POST OF THE FUTURE

5. FUNDING NUMBERS
C - F30602-99-C-0046
PE - 63760E
PR - CPOF
TA - 00
WU - 02

6. AUTHOR(S)
Amanda Vizedom

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/ IFTB
525 Brooks Road
Rome New York 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFTB
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2003-199

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Raymond A. Liuzzi/IFTB/(315) 330-3577/ Raymond.Liuzzi@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***
This report addresses: (a) Description of the Cyc Battle Monitor prototype and (b) Development of the Cyc Battle Monitor concept. The development represented by the completed Cyc Battle Monitor prototype, goes beyond recognition of surface data patterns because both the richness of the Cyc ontology and its reasoning capabilities are being utilized to provide a more robust, deeper foundation for Battlespace Monitoring. In the Cyc Battlespace Monitor concept the following are addressed: (a) What kinds of information are important to commanders in what kinds of situations? (b) To what extent is the contextual importance of information a matter of preference, (c) In a real, operational context, how, and to what extent, are the relevant kinds of information obtained, estimated, guessed, or calculated, and reported? (d) To what extent is this information currently, or feasibly, available in a format which can be read and understood by computer, (e) How are situations determined and defined, with respect to the characteristics that make types of information significant or insignificant? (f) To what extent are the situation-determining characteristics potentially available and intelligible to Cyc? (g) What information should be the output of the monitoring system? In addition among the first operational lessons learned was if technologies are needed to bring the important information out from the masses of data, and if experts work by responding to patterns, then an ideal solution should work by detecting patterns and increasing the commander's ability to see them among the noise. Many of the lessons learned - and portions of the technology produced - have already found application in subsequent DARPA programs.

14. SUBJECT TERMS
Artificial Intelligence, Knowledge Bases, Data Base, Software, Fusion

15. NUMBER OF PAGES
40

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

1. Objectives

The Command Post of the Future (CPoF) program opened in February, 1999 with the following program Goals:

- Shorten the commander's decision cycle to stay ahead of the adversary's ability to react
- Develop and experimentally validate innovative Technologies and CONOPs for the Human-System Interaction for Command and Control After Next
- 21st Century missions
- New CONOPs
- Paradigm shift in C2 = C2AN
- Derive Design Principles for C2AN HSI that will guide developers of future systems[1]

The first of these goals expanded into the following sub-goals:Increase Speed and Quality of Command DecisionsFaster recognition and better understanding of changing battlefield situation
    - Faster and more complete exploration of available courses of action
- More Effective Dissemination of Commands
    - COA capture for dissemination of commander's intent
    - Status and capability feedback from deployed operators
- Smaller, More Mobile and Agile Command StructuresFewer staff members forward
    - More mobile, distributed command element
    - Smaller support tail & reduced deployment requirements[2]

The CPoF program aimed to develop technologies which would enable a commander to use information and information technology effectively, with the result that Situation Awareness and command decisions would increase in speed and quality.   The vision of smaller and more agile future forces provided constraints on this goal: Improvements not only could not be achieved by adding staff, they had to be achieved with an assumption of reduced staff. If the technologies provided ways to bring about such staff shrinkage, so much the better.

CPoF included a diverse range of technology development groups. Some focused on the Human-Computer Interface (HCI) aspects of the above problems. Others focused on the collaborative exploration of options, and communication across a dispersed command staff. Cycorp's role in CPoF centered on the possibilities for intelligent computer processing of available battlefield information, in support of faster and better Situation Awareness; if feasible, such capabilities would result in fewer humans wading through

---

[1] Kickoff presentation, Ward Page, February 23, 1999. See CPOF home page: http://cpof.wwwhome.com/home.html.

[2] Program review, Ward Page, June 1999.

data, and improved odds of getting the Commander the information needed to make decisions in a timely fashion.

Initially, Cycorp was tasked to provide ontology support for other development branches, particularly those dealing with Course of Action capture and user task modeling aspects of HCI. The CPoF Subject Matter Experts (SMEs), however, saw the Course of Action sub-goals as distracting from the main tasks of the program and the corresponding sub-goals were dropped from the program in the middle of Year 1. The user task modeling development branches ended up taking a backseat to the HCI branches focused on collaboration. Cycorp's tasking then focused on developing knowledge-based capabilities to support the first sub-goal: faster recognition and better understanding of changing battlefield situation.

Early CPoF experiments strongly suggested the value of intelligent alerts which focus Commanders' attention on critical information at the right time.[3] Our effort followed these suggestions, and aimed to bring Cyc's representation and reasoning capabilities to bear in the production of intelligent alerts. Initially we focused on the critical areas of constraint violation and enemy intent. We conducted a series of discussions and explorations with the CPoF SMEs, and eventual selecting enemy intent as the more interesting and promising of the two areas for our prototype development.

Inspired by BGen Pat O'Neal's draft paper on Monitors, we took as our objective the develop a battlefield pattern identification and intent interpretation application, powered by the Cyc Knowledge Base and Inference Engine. Our goal was to enable Cyc to listen in on battlefield activities by reading and understanding (structured) field reports, to analyze those reports for developing patterns of activity and asset deployment, and to recognize patterns that typically telegraph important aspects of enemy intent. Finally, we aimed to go beyond identification of battlespace patterns, and into the generation of hypotheses about enemy intent.

Our aim, given the constraints of the project, was to develop a prototype that would demonstrate the feasibility of a knowledge-based Battle Monitor. We and the CPoF SMEs agreed that for this prototype stage, it was more important to develop an area of reasoning from end to end than to develop broad coverage. With this in mind, we chose to focus on patterns, and their constituent elements, relevant to Air Assault. Our task then became to develop a Battle Monitor prototype capable of picking out, from a background of other battlefield data, both relevant and irrelevant, patterns of enemy activity and deployment that likely telegraphed an enemy air assault.

[3] See, e.g., analyses of visualization experiments produced by David Noble of Evidence Based Research, Inc.

2. End-State: Description of Cyc Battle Monitor prototype and output.

After many iterations[4], desiderata for the functions of the Cyc Battle Monitor were most completely described in a pair of documents created in April of 2002: the Artillery Monitors Output Description Document and the Reconnaissance Monitors Output Description Document. For reference, these documents are included here as Appendices B and C.

We were able to complete development of Battle Monitor functions as described in parts A, B, C, and D of the Artillery Monitors desiderata document. An initial delivery of the part A output was delivered on August 21, 2002 for SME review and feedback. No feedback was received. The final delivery, for parts A, B, C and D, was delivered on January 17, 2003 to Ward Page (DARPA), Jim Shoop (ISX), Ray Luizzi (AFRL/IFTD); to our Subject Matter Experts BG Tom Garrett (IDA), MG Pat O'Neal (IDA); to SME and scenario developer Bruce Gudmundsson (Instructor, Quantico); and to our development partner at Global InfoTek, Andy Wills. This development represents the foundational ontology, inference, and application work for report interpretation, entity classification, and pattern detection for enemy artillery assets and activity. In addition to the classification of artillery assets, reasoning about their capabilities, and calculations of range and potential, the artillery pattern analysis performed on reported data falls into four categories:

- Snapshot of actual artillery fire
- Snapshot of artillery potential
- Significant changes in active artillery fire
- Significant changes in artillery potential

We would have liked very much to have been able also to fully develop the reconnaissance alerts. Most of all, we would have liked to be able to move on to combining these patterns and reasoning about what these patterns might mean in terms of enemy intent; it is this development, and the SMEs feedback on the result of it, which we most looked forward to. Unfortunately, our late tasking, knowledge acquisition challenges, the technical difficulty of some development components, as well as time, resources and coordination challenges made that task more ambitious than we were able to complete within the duration of the CPOF project. The development team (and, we're sure, the SME team) was left fervently wishing that we had been able to start this project at the beginning of the program, instead of two years in, so that we might have been able to bring it to full fruition.

[4] These iterations and the lessons learned from them are described in detail in section 3 of this report.

Nevertheless, the completed development itself constitutes a significant step forward. We have gotten a much better understanding of the operational and technical problems, and laid good foundations for further work. Moreover, the results are, in our view, interesting, and show some of the potential helpfulness of a fully-developed Cyc Battle Monitor. A discussion of the lessons learned appears in Section 3, and includes an outline of the challenges remaining.

To understand the development represented by the completed Cyc Battle Monitor prototype, it is important to understand that the accomplished behavior goes beyond recognition of surface data patterns. Rather, both the richness of the Cyc ontology and its reasoning capabilities are being utilized to provide a more robust, deeper foundation for Battlespace monitoring. The following sub-sections present more detail on the behavior of the Battle Monitor, and the capabilities it provides.

2.1 Receipt, Translation, and Storage of Battle Reports

The Battle Monitor receives battle data as individual battle reports, in the form of Java objects, from the Global InfoTek Inc. Battle Authoring Tool (BAT). These reports may be of several types: Situation Reports (SitReps) in which friendly units report on their own situation, Spot Reports (SpotReps), in which friendly troops report their observations of enemy units or activities, and GPS reports, in which a friendly unit's GPS equipment periodically sends an automated update of that unit's position. SpotReports most often come from local units; however, sources also include, e.g., JSTARS, Satellites, Unattended Ground Sensors, and HUMINT provided by upper level commands.

The reports used to drive the Battle Monitor development and testing were generated by CPOF SMEs using the BAT. The BAT enables the SMEs to simultaneously create a ground-truth data set and to create reports in which data may be omitted or erroneous. Only the reports are sent to Cyc; the Battle Monitor reasons over the reported data, and does not have access to unreported ground truth. For initial development, it was decided that the reports provided would be partial – that is, some data would be hidden – but not intentionally incorrect (it is also understood, however, that the ability to deal robustly with incorrect data is necessary for a mature system).[5]

Background information about the scenario (but not the actual battle) was provided to us before hand. The report formats were developed cooperatively by us, the SMEs, and GITI, so we were also aware of the types of information that could occur in reports.[6] Ontological Engineering was performed to extend the KB as needed, to ensure coverage of concepts needed for the representation of the scenario, forces, activities, and anything else possibly occurring in the reports or necessary for reasoning about the reports.

The specific scenario data, in particular the general composition of the forces involved and the types of equipment they were believed to possess were also represented in the knowledge base. In this case, this background intelligence was hand-entered; this task is

[5] See p.11.
[6] With some exceptions. See p. 14.

one among several which should, in a mature system, be automated; and for which we are now in the process of developing appropriate automated-access technologies.[7] In any case, at the outset of the Battle (i.e., of the Battle Monitor's receipt and input of the battle reports), the Knowledge Base (KB) includes coverage of the reportable concepts.

In its current version, the Battle Monitor translates each received report into CycL according to a hard-coded specification, based on knowledge of both the concepts in the KB and the report fields. This hard-coding is perfectly adequate to the experimental task; however, were we building another version today we would replace it with the above mentioned automated data access technologies.[8] In addition to translation, the Battle Monitor also loads the CycL report representation into the KB, creating a Report Microtheory specific to that report. The report microtheory is named using Cyc's capability to create new terms on-the-fly using functions. The resulting term consists of a function for the particular report type, and a number simply representing the report's place in the sequence of reports received: e.g., (#$CPOFSpotRepMtFn 247) is the microtheory whose contents represent the information received in the 247th spot report of the battle. This microtheory contains the reported entity and activity observations; it also features metadata like the time of the report and the source.

The units mentioned in the report are named in a similar way. For most reports, there are three possible ways in which a unit can be mentioned. Some unit will be the source of the report. Some unit will be the Subject of the information given in report. And, in some cases, some unit will be the Object of the information given in the report. In many cases, at least one of these units will be of unknown identity. Therefore, functions are also used to name the units mentioned in a report. If the actual identity of the unit is given (e.g., it is a friendly unit, or someone lucks out and reads the side of an enemy tank), then an assertion is made in the report microtheory to the effect, e.g., that the subject of that report is the enemy's 25th Armored Battalion.

Some report fields, however, are by choice not asserted into the KB. These are fields in which either observer reliability is low (e.g., echelon), or doctrinal typing is frequently misleading, or both (unit type). Instead, as each report is loaded, the Battle Monitor triggers a series of Cyc queries designed to infer the desired information from the more reliably observed fields such as equipment, approximate numbers, location, and activity.

2.2 Functional, knowledge-based classification.

The asset-type classification reasoning is based on equipment, number, and activity, identified by the SMEs as the more reliable report components. From equipment and activity, Cyc reasons about the capabilities of the unknown reported units, labeling them as likely to provide arty capability, recon capability, or transport, for example. Cyc also makes note of other significant immediate classifications; for example, certain equipment may mark an observed unit as elite (and therefore perhaps worthy of special attention).

---

[7] See discussion of Semantic Knowledge Source Integration, or SKSI, on pp. 12-13.
[8] Again, see discussion on pp. 12-13.

Importantly, the classification process avoids doctrinal unit-typing, which not only varies importantly by organization, but also typically assumes that different unit types are exclusive. It may be true that the United States Army says that you're either a Reconnaissance unit or an Artillery Unit and not both, for example, but it is also true that some units not doctrinally classified as Artillery Units carry some artillery equipment, and that just about anybody can perform reconnaissance. Thus, although the vocabulary and capability exists for each unit to represent its doctrinal unit type (and in fact this information is represented for the units we know about), the aim of the Battle Monitor classification reasoning is to infer what type of battlefield function an observed unit represents. There is no assumption that these functions are mutually exclusive.

Knowledge about groupings of equipment, e.g., that a particular arty control vehicle is used in conjunction with a particular arty launch vehicle, is used also, e.g., to infer the presence of the launch vehicle from the observation of the control vehicle.

We made some progress toward the goal of further utilizing unit location (relative to other units, relative to Forward Line of Troops, etc.) in classification. We were not, however, able to make sufficient progress within the time constraints to include this capability in the CPOF prototype. [9]

2.3 Entity Fusion and Knowledge Updating.

An initial pass at entity fusion was included; as each report is processed, the Battle Monitor also triggers Cyc queries designed to discover whether the entities mentioned in this report are the same as those in some previous report. The level of fusion developed sufficiently handles the most accessible cases, e.g., named units, and units with identical or subsumed characteristics in identical positions. This first-pass fusion supplies information for report updating. For example, if a friendly unit reports its location and situation more than once, with each subsequent report, the fusion reasoning identifies that this report subject is identical to the report of a previous subject, and that the information contained in the new report supercedes previous information of the same sort. [10] The principal significance of entity fusion, and the reason we would like to do more of it, is that fusion makes the difference between thinking you've got four artillery units massed on a location and realizing that you've got only one, reported four times. For a mature system to be useful, entity fusion must be a well-developed part of it.

2.4 Echelon reasoning.

During the report-processing stage, the Battle Monitor also triggers a series of Cyc queries aimed at discovering the likely echelon of the assets. The echelon reasoning is based on types of equipment (e.g., using the intel about the opposing force to note the echelon at which MRLS assets are directed).

---

[9] See discussion on p. 12.

[10] There are many possibilities for more sophisticated fusion and updating. For some discussion of those we have in mind but have not yet developed, see discussion on p. 12.

The principal significance of this reasoning is the detection of higher-echelon assets. The disposition of such assets is sufficiently indicative of enemy focus to warrant its own variation on each asset-type pattern; for example, under SME guidance we included queries to discover specifically the location of, and massings of, division-level artillery only.

2.5 Knowledge-based reasoning about effects: ranges and capabilities appropriate to classification results.

Additional Cyc queries are triggered to figure out the ranges and capabilities of units once they are classified as arty-capable. This reasoning features integrated use of background domain knowledge, force intelligence, computational modules, and external map data.

2.5.1 Artillery Range

When units are assessed to be arty-capable, Cyc uses report assertions about their equipment possessed (and background intel, if any), plus background domain knowledge about the characteristics of particular types of weapons, to find the maximum-range weapon the unit possesses and, from there, the unit's maximum artillery firing range. This range is combined with location data, externally stored map data for the battlespace, and distance calculations to figure specific ranges of fire on the map. Finally, massings of artillery potential are found by calculating overlaps in these ranges. These massing sets are re-evaluated as time advances, whenever there is new artillery information.

A computational geometry module was developed for this purpose, so that the range and massing information could be calculated efficiently. The architecture of the module, however, is not specific to artillery. Rather, it was designed with additional uses specifically in mind, such as the determination of overlaps of reconnaissance, travel, or common ranges of any type.

2.5.2 Artillery Potential

When units are assessed to be arty-capable, Cyc also uses report assertions about their equipment possessed (and background intel, if any), plus background domain knowledge about the characteristics of particular types of weapons, to find the firing capacity of each weapon possessed and to figure total capacity for the unit. The firing potential of artillery equipment is represented uniformly in the KB using an M77-Bomblets-per-minute measure. This allows the comparison of artillery potential across units, comparison of the active artillery to the total possessed by a unit, or, in combination with the range reasoning described above, to reason about the total potential in range of a target, or massed on a location, and to compare the potential massed on an area to that massed elsewhere, or to that available in the battlespace overall.

Reported levels of incoming fire are not exact, but rather in rough ranges of High, Med, and Low. These are also converted into rough ranges of bomblets-per-minute, and compared to the potential massed on a region to determine how much of a force's potential on an area they are actually using there.

2.6 Reasoning about changes over time.

The snapshots of artillery firing and potential are in some cases interesting, but what is most telling to a commander are the changes in those patterns over time. This is the significance of parts C and D. For each of the kinds of patterns Cyc reasons about at each step in the battle, it also compares these patterns with previous results. Changes produce additional alerts, with comparatively more urgent text (e.g., "NEW MASSING!" "INCREASED MASSING!"). Some significant changes for which the Battle Monitor queried include increases in the enemy artillery massing over some general region, and increases in the amount of enemy artillery firing on some location. When changes over a particular threshold are detected, appropriate change alerts are produced. The contents of change alerts include, for example, the new percentages, the percentage change that a new massing represents compared to what was previously detected in the area, or any inferred change of echelon.[11]

A concrete example, and a reminder of how to view the results via the BAT, integrated with the battle data and reports, is provided below under "What you'll see" and "Instructions on loading and viewing" in Appendix A.

2.7 Archive of final result.

Archived final result: We will also archive the Cyc image, battle files, and Battle Monitor output, and burn this to CD. Opening the battle files and Battle Monitor output in the BAT enables viewing of the results at any time, and in context. The BAT integrates the battle files (ground truth and reports) and Cyc alerts by time, and allows one to play through the battle, observing what is happening, what is being reported, and what Cyc is concluding at each point.

Unfortunately, the interface between the BAT and the Battle Monitor is not sufficiently tight to allow access to the underlying Cyc reasoning through the BAT interface. This is why we are also including the archived Cyc image in which the Battle Monitor asserted report data and on which the Battle Monitor ran its queries, in the state it had reached at the end of the Battle Monitor run. Though an extra step is necessary here, this makes it possible to view the inferred patterns and classifications within the KB Browser and therefore to query for their justifications. The reasoning used by Cyc during the processing of the battle data is therefore available along with the results of that reasoning.

---

[11] For a relevant discussion of some temporal reasoning challenges faced at the time for which we have since developed solutions, see discussion on p.14.

3. Development of Cyc Battle Monitor concept.

Knowledge Acquisition (KA), development of requirements specifications, and general research on the Cyc Battle Monitor development effort began in early 2001.[12] This direction was inspired by BG O'Neal's "Monitors" paper-in-progress and by related discussions at CPOF work sessions. We went through several iterations of intensive KA with the SMEs, trying to get sufficient understanding of the desired behavior, the informal use-cases the SMEs had in mind, and the expert and general knowledge that could underlie the type of monitor system they envisioned. Such KA goals included answers to the following questions:

- What kinds of information are important to commanders in what kinds of situations?
- To what extent is the contextual importance of information a matter of preference, and to what extent are there fundamentals with which we might work? If there are fundamentals, what are they?
- In a real, operational context, how, and to what extent, are the relevant kinds of information obtained, estimated, guessed, or calculated, and reported? With what accuracy?
- To what extent is this information currently, or feasibly, available in a format which can be read and understood by computer, and therefore potentially available as input to Cyc reasoning?
- How are situations determined and defined, with respect to the characteristics that make types of information significant or insignificant? What distinctions can we get a handle on here?
- To what extent are the situation-determining characteristics potentially available and intelligible to Cyc?
- What information should be the output of this monitoring system? What should be the content, form, and frequency of an alert? How variable should this be? What makes an alert useful, useless or even detrimental to a commanders' ability to understand the battlefield situation in a timely and accurate manner?

We landed, eventually, on the idea of focusing on "indicator" activities – those enemy activities which often constitute pattern-recognition triggers for friendly commanders, or indicators of enemy intent. The idea behind this approach is that indicator patterns are often composed of many, dispersed events and facts about the battlespace, not significant in isolation, and difficult for any one human to track, or to pick out amongst the many other simultaneous events and facts. Cyc, however, has capabilities that make it a good candidate to reason about such data points (dispersed events and facts about the battlespace) and the connections between them, to track those facts and connections as they change, and to infer whether those data constitute or suggest indicator patterns that should be brought to the commander's attention.

[12] Regarding this late tasking, see discussion on pp. 15-17.

Once we focused on the goal of reasoning about observables (i.e., things which are or might be actually reported in field reports, with some regularity and reliability) and monitoring for indicator patterns, we began supporting ontology work and redirected our KA. Now, our KA goal was to extract from the SMEs sufficient information to identify specific indicators patterns, to define those patterns well enough to distinguish them from obvious and uninteresting similar situations, and to learn how those patterns might be discovered from observable (and reportable) data.

Some candidate indicators (e.g., subtle changes in spatial configuration and distribution over terrain and with respect to opposing forces) were excluded on the grounds that they were probably out of range for the CPOF development time and funds remaining (about a year of full time effort for existing staff).[13] Some were excluded on the grounds that the data composing the pattern were either not ordinarily observable or not ordinarily reportable (or, more specifically, not reportable within the standard range of formatted field reports in use in, or in consideration for, the CPOF experimental framework).

Others candidate indicators (e.g., massing of artillery potential, or changes therein), were selected as potentially highly useful to commanders, within range for this effort, and inferable from data that could be observed and reported. KA sessions were then dedicated to breaking the indicator patterns down into their potential components, and gathering sufficient information to enable the inference of those components from observable, reportable data. Ontology development was focused on formalizing that knowledge and enabling such inference.

[13] Regarding the potential for, and challenges of, analyzing spatial configuration and distribution, see discussion on p. 12.

4. Lessons Learned

A variety of non-trivial lessons emerged from the CPOF effort, some of which we were able to act on nearly immediately, and some of which presented longer-term challenges. This section includes reflections on the lessons learned, divided into three categories: the operational, the technological, and the methodological/programmatic.

4.1 Operational Lessons

The CPOF program featured intensive sessions with SMEs, known as Command and Control University, or C2U. The points made during these sessions were many and varied, and were brought home by a range of thought experiments and simulations. Much of the emphasis of these exercises was on interface, visualization, and collaboration difficulties and tools, and while we paid attention to and appreciated these lessons, they did not bear directly on our area of technology, and so will not be discussed here.

On the other hand, among the first operational lessons emphasized by the CPOF SMEs was that of operational overload. We learned, from their discussions and recommended materials, much about the deep need for technologies which help to cut through this overload. We also learned of the very different nature of SME decision making from what we had supposed it to be. The importance of pattern recognition and intuitive response to expert decision making is not something we had previously understood, but is one which we came to appreciate seriously.

These two lessons came together in an interesting way. If technologies are needed to bring the important information out from the masses of data, and if experts work by responding to patterns, then an ideal solution should work by detecting patterns and increasing the commander's ability to see them amongst the noise.

On the other hand, we came to appreciate the cognitive demands on the commander and command staff. The right balance between offering too many items for attention and two few, between being too intrusive and not intrusive enough, is not obvious. The differing opinions and styles of the SMEs also made clear to us that this balance is not constant or uniform. Another requirement on an operationally useful system emerged from this: it must be, in many ways, user-customizable. It must allow command staff to tailor its behavior to their needs and preferences.

This tailoring, moreover, must be intuitive and fairly straightforward. It cannot require the calling in of technical experts, given shrinking staffs and time constraints. Neither can it require the commander to break out of his or her natural way of thinking and communicating. Some training requirements may be unavoidable, but once trained, a commander should be able to communicate with the system without switching into a different mindset, a special language, or an unnatural level of precision. The cost of such switching is too high; the commander's focus on the situation and ability to develop intuitions about it, are of extremely high importance. Useful technologies should not require them to break out of the very mode of expert thinking that make them so valuable.

Additionally, we learned much about the quality of data typical of operational settings. That there are limits to the knowledge one can expect ground forces to have, never mind to report, was demonstrated forcefully. That any operationally useful system must be able to work with these limits was also something we came to appreciate. This is a level of robustness not easily obtained, but it is one which a common-sense-based system like Cyc ought to be able to eventually provide. When data is frequently bad or confused, recognition will never suffice; meta-reasoning about the data, data sources, conditions of observation, and other contextual elements will be needed. We thus came to understand this as part of the challenge as well.

Interface challenges were mentioned briefly above, with the note that they would not be significantly discussed here. Two points arising from those discussions and experiments, however, are worth noting here. We came to appreciate that operationally useful systems must be very fast (compared, say, to strategic level analysis tools deployed in a domestic office setting), and must be useable on small, lightweight devices compatible with the mobility and activities required of a deployed command staff. This underscores some of the efficiency issues discussed in next section. While Cyc-based systems are improving constantly, efficiency remains an area of continuous challenge.

4.2 Technological Lessons

Once the goal of a battle monitoring system had been set, and once we (the Cyc development team, SMEs, and DARPA program manager) focused on creating the reasoning and application code to monitor battle data for indicators of enemy intent, we faced a number of technical challenges. Some of these we were able to develop solutions within the limited development time available. For some, we were able to identify and initiate the development of solutions, but were not able to complete them within the duration of CPOF. Some remain outstanding, though we have thoughts on how we might solve them, and what is needed to do so.

Challenges involved in intelligent entity classification and reasoning about capabilities are continuously troublesome for traditional, hard-coded monitoring and/or fusion programs. These, however, were perhaps the easiest for us to solve. The expressiveness of the CycL language, and the qualitative and flexible nature of Cyc reasoning, brought these problems well within scope. Similarly, feature-based elements of entity fusion turned out to be easiest for us, where they are hardest for traditional (e.g. Bayesian) systems.

However, other elements of entity fusion and capability reasoning required significant mathematical / geographical computation. These functions can be performed in SubLisp, Cyc's underlying inference language, but are far from efficient. Meeting these challenges required building or connecting to modules designed for such mathematical work, and we did so. This allowed us not only to perform the range and intersection reasoning described in section 1, but advanced Cyc's ability to perform similar calculations as part of general reasoning more broadly.

In fact, the range-intersection and massing reasoning also required the integration of entity knowledge and map data. After all, knowing individual ranges does one little good if one cannot find the entity locations and compute the relationship between those locations. However, representing each map point, e.g., in the Cyc KB would be infeasible and inefficient, in both ontologists' time and the sheer quantity of assertions that would then require storage. Instead, this problem was solved by building an interface to an open-source GIS program, OpenMap, and developing MapServer code which kept an update map-side model of entity locations, appropriately connected to those entities as represented in the KB.

The entity fusion problem contained another problem, however, one which we did not properly anticipate. To properly reason about the possibility that two unknown entities are in fact identical, movement constraints must be taken into consideration. Some factors going into movement reasoning are at our disposal, e.g., speeds of vehicles over various kinds of terrain, reported equipment data, and so forth. So far, so good. However, the terrain data itself was not so available. Some of the relevant terrain data was included in the data set provided, and some was not. However, even if we had the remaining terrain detail, and sufficient time to tackle it, we had another problem: While we can now effectively extract point-based information from GIS data, we had not yet solved the problem of effectively extracting line- and polyline-based information. While we have no reason to think this solution impossible, we haven't yet developed it. So while we can calculate the straight line distance between two points, and while we can find the slowest known equipment for two units and thus the max speed for each, and while we can figure in the time between reports, this reasoning about movement constraints does not take into account real terrain features and variability, nor effective path reasoning. This unanticipated technical challenge, therefore, remains.

Another lesson emerged from the report-translation process. For our prototype, the ability to translate and input of data was accomplished by programming field-meaning directly into the Battle Monitor translation code. This is not a long-term solution, for several reasons. First, field meanings change, and requiring a programmer to rewrite and recompile the code each time something changes is not reasonable. Fields and field meanings should be changeable by single, simple entry actions, ideally by the user, and certainly without requiring recompilation. Second, field meanings, as used and understood by users, are not as crisp and clear as the limitations of the procedural language require; Cyc's expressiveness, flexibility, and ability to recognize special conditions is needed here, and should be accessed in this input process.

This latter challenge was one we recognized early on. The issue of efficient access to information outside of the KB is one that was often discussed, and we also recognized that we had a technical proposal in-house that presented a potential solution to our Battle data access problem. That proposal was for a new area of generalized infrastructure within Cyc, a Semantic Knowledge Source Integration (SKSI) capability. We also recognized that we did not have the time or the resources, under CPOF, to address this possibility. The Battle Monitor development experience, however, lead us to understand the importance of SKSI for BattleSpace reasoning tools in general, and changed the priority and focus of this technical direction. In fact, if at the outset of CPOF we had this understanding of the special relevance of SKSI in this domain, we would have proposed its development as our primary role in the program.

An additional challenge arose from the volume of data we were translating and loading into Cyc. We were unpleasantly surprised to discover that when we got to a certain point in a run through the battle data, processing would slow down significantly, and eventually stop altogether. Presently, all knowledge in the KB resides in memory when the KB is loaded. Adding thousands, if not tens of thousands, of new assertions during a continuous KB run therefore presents a significant problem, and one we soon ran into: memory limitations quickly slowed the Battle Monitor down to an unacceptably slow speed, and eventually would stop it entirely.

Again, the possibility that memory limitations would soon present Cyc with significant problems had been discussed, and we discovered that we had an in-house technical proposal to develop a KB Backing Store for Cyc, with the appropriate Inference Engine hook-ups, so that the entire KB would not be in memory all time. The technical proposal existed because this development was recognized by some of our senior developers as absolutely essential for Cyc's development; else, at some point down the road, there would be a scalability failure. However, because of our limited experience with near-real-time, massive-data applications, we had not understood the consequences there, nor how quickly they would create a problem.

Again, though, we did not have the time or the resources, under CPOF, to address the needed development. The Battle Monitor development process, however, led us to understand the connection between this technical issue and our efforts to build useful reasoning tools for the military domain. The priority and focus of this issue also changed.

Since we did not have either the resources or the time to develop these solutions within CPOF, but now recognized their relevance to the military domain and their urgency, we developed a SBIR proposal based on our CPOF experience. That proposal outlined the operational problems we were attempting to address in CPOF, the problems faced without something like SKSI and the Backing store, and the possibilities provided by those technical directions. We were granted the SBIR I, in which we analyzed more closely whether the developments in question would really be likely to work, and to solve our problems. We were then granted the SBIR II, and that work is under way and making excellent progress. Even now enough has been done to have helped us develop more efficient access to the battle data. Already, were we starting over on the Battle Monitor next iteration, we would be able to substantially improve its design and efficiency.

We also learned some lessons about integration between interface and application. When we received our Battle Monitor tasking, GITI received parallel tasking to support us by building both a report-entering tool with sufficient knowledge-engineering foundations to deliver sufficiently meaningful data, and to develop an interface for the display of Cyc output.

The first of these we knew by this time to be crucial. We knew this because we had attempted to make use of the reports being created for the CPOF visualization and collaboration experiments and found that the format of these reports introduced such ambiguity and confusion into the reports that they were unusable for machine reasoning. Since by this time CPOF was heavily focused on, and invested in, the visualization and collaboration technologies, this format was not going to change. Therefore, the decision was made to task the SMEs to provide us with useable battle reports, and GITI to provide the SMEs with a tool for doing so.

The second level of integration we also knew to be necessary, because the natural output of Cyc is not generally readable by persons other than Cyc technologists, and is certainly not in a form to be informative to a commander in-battle. The integration, however was only partial; it was really more a matter of maintaining parallel domain models in both the BAT development and Battle Monitor development universes. This maintenance failed on a number of occasions when, e.g., a visiting SME would make changes to the BAT report format or values, or aspects of the scenario would change. Furthermore, the process of getting feedback from the SMEs on Battle Monitor development required that the two always be in step, but the relative independence of the two applications (and the heavy additional programmatic tasking of GITI) meant that this too often failed, making it difficult to get SME feedback on small steps in Battle Monitor development. Also, features and information available on the Cyc side was not necessarily accessible to the users; e.g., while justifications of inferences are always available in Cyc, a user could not click through the interface to get at them. And finally, even at the level of integration we did accomplish, a great deal of time and energy was spent keeping the two development teams on approximately the same page, and in each helping the other understand their capabilities and limitations. A lesson we took away from this is that, first and foremost, though it is easy to overlook or view as trivial, the question of interface and integration must be given serious time and resources. An additional lesson is that the user-desired level of access to underlying functionality must be part of the specification from the beginning, so that the appropriate level of integration can be designed and planned into scheduling and development.

Finally, as mentioned earlier, some of the biggest challenges for Cyc-based applications lie in the way of efficiency. Many steps, including the SBIR-funded effort mentioned above, are underway, and more are in the queue or on the drawing board to dramatically improve the speed of Cyc reasoning even while increasing its depth, it sophistication, and the breadth of knowledge on which it draws. Recent changes in the fundamental architecture of the inference engine have made many such improvements possible, such as the new, much improved handling of temporal reasoning – a development that could simplify and improve several aspects of Battle Monitor performance, including the detection of significant changes over time. Some of these changes have been driven significantly by the lessons of the CPOF effort; the development of dramatically improved temporal reasoning, for example, was partially funded out of our CPOF effort until it became clear that it would not be completed in time to deploy before the end of the program.

Given the challenges described above and the limited development time remaining once we got our Battle Monitor tasking, the real operational payoff is of course not yet there. That payoff requires, in our opinion, more complete coverage of battlefield events and entities, and especially of interesting higher level patterns that might occur across a more full range of phenomena. It also remains to further develop solutions some technical challenges, regarding especially the speed of reasoning and the efficiency of information storage.

4.3 <u>Methodological/Programmatic Lessons</u>

4.3.1 Importance of data selection/design.

The Battle Monitor development process was slowed down immensely by the fact that it started without either experimental data or an initial set of requirements. An immense amount of developer time and funding was spent in the effort to procure these two fundamental drivers of technology development

The battle data used to drive the Battle Monitor was created by CPOF SMEs, simulating a battle over given terrain and with given forces, and simulating the reports that might be generated during this battle. This process is, understandably, slow and tedious for the SMEs. The Battle Authoring Tool was largely intended as a tool for the easier, faster generation of such reports. It includes facilities for the creation of the main forces and units before hand. As the SMEs create new reports during the battle, the units they select and type of report they indicate cause some data to be automatically filled in to the report fields using the unit profiles. The SMEs can then modify these if they wish to simulate erroneous reports, or obfuscate portions of the data so that they will be included in the ground truth battle data, but will not be part of reports sent out to Cyc (or to any other client application). Nevertheless, creation of realistic battle data remained an obstacle during CPOF, given limited time and even more limited SME resources.

The format of the reports was developed carefully. The foremost consideration was the plausibility of field unit's having particular information and/or the feasibility of their being able to report it. In some cases, fields were made available but expected to be used rarely if ever during the battle (e.g., precise identity of enemy unit). In other cases, fields currently in use were included, but their use by Cyc was limited by SME assessments of their reliability (e.g., echelon of enemy unit, as estimated by friendly observers).

The second ranking consideration was the clarity of the meaning of the data. Report formats lacking careful knowledge engineering can introduce new ambiguities into command communications, not present in, e.g. voice communications. This lesson was brought home the hard way, as we attempt to make use of data produced for MayaViz experiments. This experience, in turn, leads to another, methodological lesson: Experiments custom-designed for and entirely run with a focus on, a particular technology are not necessarily going to be of use for a different set of technologies.

4.3.2 Understanding the limitations of shared program resources.

This leads to a second methodological lesson. The small number of experiment-support personnel, and especially the small number of SMEs, contrasted significantly to the large number of technology development groups in the program. This meant that some groups got little to no support. In our case, it meant that we got almost no experimental support, and got SME support only late in the game (i.e., two years into the program). Much wasted time could have been avoided if this were understood up-front, and/or if there were a programmatic plan for the relationship between the different development groups, such that it was understood that some would have to work off of the efforts of others.

4.3.3 Knowledge Acquisition and Technologist-SME Interactions

Much of our KA effort was spent in technologists' development and offering of descriptions of candidate systems and monitor concepts for critique and expansion. While we started from BG O'Neal's paper, we quickly found that past that point, the SME's found it extremely difficult to describe what they wanted. We did not have independent programmatic support for SME development of use cases, nor did the SMEs have support or practice in such development. Without use cases, and without sufficient time or knowledge[14] for Wizard of OZ (WOZ) prototypes, we found that offering descriptions of potential monitors, along with candidate rules (in English, extracted from SME discussions and SME-recommended reading materials) and reasoning, was the best mechanism we had for eliciting requirements and specifications.

After a certain degree of refinement, however, the SMEs found such descriptions confusing and/or insufficient for them to know whether the application might be useful. At times they expressed this confusion or insufficiency; more often we discovered it via multiple cases of our getting contradictory answers to questions, or contradictory responses to essentially the same proposal at different times. We concluded, after some

[14] Or desire. Throw away code is frustrating, and especially difficult in the case of KB extension, where the development is integrated into, and may uncover and require needed corrections to existing KB content that changes the behavior of portions of the KB itself.

It may be argued that Throw –away code can still be developed that offers a Wizard of OZ work-around for any such knowledge, in procedural application code such as direct Java interface programming. This is only partially true. It is partially false because it misses the very reason the project is a Cyc project; Often the behavior that is desired is sufficiently rich, complex and contextual that it cannot be mimicked easily, or at all, by simple procedural code. The development of, e.g., Java WOZ mockups still requires sufficient KA to develop an initial specification, and then an attempt to capture that specification in a mock-up demonstration. The training of the Cycorp staff, however, is overwhelming in the logical and technological areas and languages required to develop knowledge and inference in Cyc, and in some cases in the kind of application writing that utilizes Cyc. It is, by design, not much in the area of writing the kind of mock-up demos in questions. The CPOF development team, like most of Cycorp, was staffed by technologists ready and able to develop real Cyc capabilities. We did not have, nor could we have easily borrowed, staff trained in the production of throw-away demo-code, even if the specifications we were able to elicit had been of a sort that could be mimicked by such code with a shorter development cycle than actual Cyc and application development.

time attempting to improve descriptive clarity, that no further clarity was going to be achieved without giving the SMEs the initial application to which they could react.

This continued to be the case with future iterations. The raw output of the inference engine, or even such output translated into English, is largely unintelligible to the SMEs. This was expected, and part of the reason for the designed-in goal of presenting this output visually, with a graphic-text combination, in the GITI BAT. However, this led to two complications. First, each development and feedback cycle had to include not only the development of the reasoning, but also the development of the required interpretation and display mechanisms in the BAT, and the re-synching of the Cyc representation and the BAT's, the latter of which was at times altered by revisions by other SMEs and not necessarily communicated to the Cyc Team or in keeping with the underlying knowledge engineering principles. Since the GITI team themselves were tasked with not only this support but multiple other CPOF efforts, these additional needs sometimes introduced not only delays, but additional confusion and technical hitches.

4.3.4 Importance of Programmatic Clarity and Requirements

All three of the points above relate in part to a single issue: while the vision of the CPOF program was clear and strong, its technological goals were vague at best, contradictory at worst. Almost none of the development groups had clear tasking from the get-go. This was acknowledged, and the program leaders expressed a desire to let technological capability and operator expertise design the requirements. At first this approach was welcome, a refreshing change from some other overly-specified, insufficiently-informed projects. But no framework emerged, and all development teams were continuously looking for something to build off of, some set of data, some set of situations, some set of focal technology. For almost every case, serious development could not begin without some such frame of reference.

We worked with other development groups (e.g., the dialogue, context, and COA working groups) to develop some framework, but in each case, while developer agreement was possible, we all lacked the data or specifications from which to start. Even the group facilitators seemed to have little idea where we should be headed.

Each of us, therefore, spent time working on our core technologies, particularly on the aspects which seemed most likely to apply to our eventual tasking, and on the aspects listed in our original Statements of Work. Program plans and future experiments would be announced, and we would refocus the work to support them (e.g., expanding KB coverage to support an upcoming Tactical Decision Game), only to discover in a few months that those plans had been dropped, and new plans replaced them.

The point here is not to grumble about what was, in our opinion, a well-intentioned and rather bold methodological experiment. The point is a methodological lesson: specifications do not emerge from nowhere, and without at least early data sets or early functional requirements, little is likely to get done, even with all the educational sessions and group brainstorming sessions in the world. This is especially true for development teams without significant SME access. Were we to be in the same position today, we would push early and often for one-on-one sessions with SMEs of the sort we began in 2001. Had we gotten to the same tasking a couple of years earlier, we believe we could have completed a Battle Monitor prototype including the full list of desiderata, and demonstrating true reasoning about enemy intent. Were we to find that such access was not possible, we would push early and often for focal data sets and/or operator-created use cases, or at least system behavior specifications at some minimal level of detail and specificity.

In summary, then, we accomplished a great deal, and delivered a working prototype system, as the task itself evolved. We hope that this work will prove useful to those coming after us, to operationalized the technology and/or to do further R&D extending the technology that has been developed under this contract. In fact, many of the lessons learned – and portions of the technology produced – have already found application in subsequent DARPA programs in IXO and IAO.

Appendix A: Instructions for Loading and Viewing Battle Monitor output.

<u>Instructions for loading and viewing via the BAT</u>

1. Start the BAT.
2. Select "Open" under the "File" pull down menu.
3. In the "Open" window, find the most recent bluegrass battle file (ends in .xbf). This should load the battle that the generals scripted.

4. Again under the "File" pull down menu, go to the "Import" sub menu and select "Inferences". This will bring up another "Open" window.

5. At the bottom of the open window, click "Files of type" and choose "All Files (*.*)"

6. Find the Cyc output vector.object file.

7. Now you've loaded the battle file and the Cyc inference file. Next, look on the toolbar and click on the Cycorp logo (sixth button from the left of the toolbar). This should bring up a "Cyc Inferences" window.

8. Now press the play button on the upper right hand corner. Now the battle is off and running at a rate of 1min game time per 1sec real time. Cyc inferences will appear in their window in sequence with the game time.

<u>What you'll see when you view the results:</u>

Ranges associated with firing alerts appear in red.

Those associated with alerts regarding massing of potential appear in yellow.

When a report is clicked on in the alert table (the left-hand side of the Cyc Inferences window), these ranges appear on the map, and the text alert messages appear in the right side of the Cyc Inferences window.

These text alert messages expand when clicked upon, and cause the appropriate ranges to be highlighted and labeled on the map.

<u>Example:</u>

At H+0d:01:50m11s a new alert appears in the alert table, with the message "Artillery Potential Alert."

Clicking on this alert in the table causes a number of artillery ranges to appear on the map, colored yellow. Each is also labeled with the reported unit whose range it is, although this is usually an unknown unit labeled as, e.g., the subject of SpotRep 267.

On the right, a folder appears labeled “Cyc Results.” Underneath it appear additional folder for the various levels of massing recognized here. In this particular example, there are two additional folders, labeled “Division level artillery” and “Brigade level artillery.” Clicking on the icon next to either gives you more information about that level.

In this example, clicking on Division level icon produces another with the message “Percentage of total potential firepower in massed region:

90.5%. Clicking on this in turn produces a list of all units contributing to the massing, and another icon labeled “Targets in massed region.” Clicking on this targets icon produces a list of all known potential targets within the massed-on region—in this case, a variety of South Bluegrass units.

Clicking on the Brigade level icon, in this case, produces a message about a change in massing: and icon labeled “INCREASED MASSING!” This is accompanied by the percentage of total firepower massed here, 11.5%, and the percentage increase from the last report, 21.05%, and the previous highest echelon with the area in range. Clicking on the increased massing icon, as in the above case, produces a list of the units involved in the massing, and an expandable icon for a list of known targets in the massed-on region.

In addition, clicking on each of these sub items in results causes only the ranges pertinent to that particular sub item to be highlighted on the map.

Appendix B: Annotated Artillery Monitors Output Description Document

Note that the schedule of work described here depended on our have the BAT data as specified, and as the SMEs intended to provide it. However, due to BAT use problems and due especially to conflicting CPOF tasking of GITI and limited SME time and response availability (for, we assume, partially the same reason), the corrected and full battle data (i.e., including locations of reporting friendly units) was not received by us until July, well after the arty work was to have been completed. This pushed the schedule back more than proportionally, as by the end of July most of our CPOF staff were committed to move to – and needed on – other projects.

Schedule as of April, 2002 meeting at Shafer Corp/ ISX:

- Enriched BAT data: Andy will try defogging everything. Remainder needs will be addressed by Tom and Pat. If all goes well, 5/3/02 delivery.
- **A&B** below: output delivered to SMEs on 5/7/02.
- GITI and ISX get latest BAT onto SME machines so they can review the above: 5/7/02.
- **C&D** below: testing complete 5/30/02, output delivered to SMEs on 6/1/02.
- **E&F** below: testing complete 5/30/02, output delivered to SMEs on 6/1/02. Note has been made that we have some worries about this delivery date. We'll try to make it; the possibility of delay is acknowledged. E&F, however, are agreed to be of lower immediate importance. SMEs want custom monitors with a customization interface eventually, but for now want demonstration of the reasoning and its usefulness.

Assumptions:

- Battle data will be in current format provided by GITI BAT. If new scenario cannot be gotten in that format, we will stick with current scenario. Default: New scenario data will not be available any time soon. Ward hopes to have new scenario data out of June experiment. At least until then, we should continue to use the BAT scenario.
- Per Ward: Same, comprehensive set of queries will be run every time. Comprehensive output object will be passed to GITI every time. We are *not* worrying about when what should be displayed anymore. That selection / activation functionality will be built at the tool/interface level. Our job is just to provide all the answers in case they are wanted at any time. May want to add a flag in cases that are more naturally alerts, vs. query answers. But likelihood is that as a placeholder, GITI will just make a clickable folder for each question, or something like that.

- Evaluations/demos will not be real-time. Battle will be run through Cyc Battle Monitor to produce comprehensive output, which will then be selectively playable through BAT.
- Questions should be run over all forces. We will do this in separate runs, and produce separate output objects for each force (considerations in favor of this include time to run, size of output).
- Relevant to reasoning about echelon of arty assets actually firing: at this point we are getting information about the type of munitions or guns/launchers involved in only a few reports. The majority of reports of arty fire that we get don't have these fire details. We can infer echelon in just the few cases where such info is present; otherwise we really don't have anything to work with.
- Andy will check on fogging. If this doesn't resolve it, he'll get new BAT and pointers (with help from Alan) to Tom to fix. Same with missing locs on overhead recon reports, and reports with NB sources.

Content

I) **<u>Questions for which answers will be provided at each time click (English).</u>**

Note: no suggestion is made here regarding what should be displayed how, or in what order. The below specifies only what information will be available in answer to what question; we assume (as we understand Ward intends) that the question of how to display it is better left to those developing the interface. Also, phrasing of questions and descriptive strings is easily changeable to make things more clear.

A) <u>Snapshot of arty potential</u>

1 *Where is the arty potential concentrated?*

Answer returned: list of all massings, info for each in format IIA (see section II below for general format descriptions):

a string: "Massed arty potential"

b list: lat-long and radius (max range) for each unit involved

c number: percentage: total potential in this massing / total potential on board

d string: highest echelon assets involved in this massing {"Co"/"Bn"/"Bde"/"Div"}

e list of pairs: possible targets (known objects of interest) within the massed-on region. For each:

i string: name or descriptor for known object

ii lat-long

2 *Where is the arty potential <u>most</u> concentrated?*

Answer returned: list of the single massing, or multiple massings in case of a tie, involving the highest percentage of arty on board, info in format IIA:

a string: "Proportionately greatest massing of arty potential"

b list: lat-long and radius (max range) for each unit involved

c number: percentage: total potential in this massing / total potential on board
d string: highest echelon assets involved in this massing {"Co"/"Bn"/"Bde"/"Div"}
e list of pairs: possible targets (known objects of interest) within the massed-on region. For each:
  i string: name or descriptor for known object
  ii lat-long

3 *Where is the {ech} arty potential concentrated?*
Answer returned: list of all {ech} level massings, info for each in format IIA:
a string: "Massed echelon-level arty potential"
b list: lat-long and radius (max range) for each unit involved in this massing
c number: percentage: total potential in this massing / total potential on board
d string: highest echelon assets involved in this massing {"Co"/"Bn"/"Bde"/"Div"}
e list of pairs: possible targets (known objects of interest) within the massed-on region. For each:
  i string: name or descriptor for known object
  ii lat-long

4 *Where is the {ech} arty potential most concentrated?*
Answer returned: list of the single massing, or multiple massings in case of a tie, involving the highest percentage of {ech} level arty on board, info in format IIA.
a string: "Proportionately greatest massing of echelon-level arty potential"
b list: lat-long and radius (max range) for each unit involved
c number: percentage: total potential in this massing / total potential on board
d string: highest echelon assets involved in this massing {"Co"/"Bn"/"Bde"/"Div"}
e list of pairs: possible targets (known objects of interest) within the massed-on region. For each:
  i string: name or descriptor for known object
  ii lat-long

B) <u>Snapshot of arty fire</u>

1 *Where is the arty fire?*
Answer returned: list of current targets of arty fire. Info for each in format IIB.
a string: "Target of arty fire"
b lat-long of location receiving fire.
c number: percentage: active fire on this location / potential able to hit this location

d number: percentage: active fire on this location / total active fire on board.
e string: highest echelon assets involved in firing on location. [*see assumptions]
f list of pairs: possible targets (known objects of interest) at (within specified distance of?) the targeted location.
   i string: name or descriptor for known object
   ii lat-long

2 *Where, proportionately, is the arty fire greatest?*
Answer returned: list of single, or multiple in the case of a tie (probably frequent, given the coarseness with which firing volume is specified), target(s) taking the highest percentage of arty fire (fire on that location as a percentage of total known fire in battlespace). Info for target(s) in format IIB.
a string: "Target of highest percentage of arty fire"
b lat-long of location receiving fire.
c number: percentage: active fire on this location / potential able to hit this location
d number: percentage: active fire on this location / total active fire on board.
e string: highest echelon assets involved in firing on location. [*see assumptions]
f list of pairs: possible targets (known objects of interest) at (within specified distance of?) the targeted location.
   i string: name or descriptor for known object
   ii lat-long

3 *Where is the {ech} arty fire?*
Answer returned: list of current targets of fire from {ech}-level arty assets. Info for each in format IIB. [*see assumptions]
a string: "Target of echelon-level arty fire"
b lat-long of location receiving fire.
c number: percentage: active fire on this location / potential able to hit this location
d number: percentage: active fire on this location / total active fire on board.
e string: {ech}: highest echelon assets involved in firing on location. [*see assumptions]
f list of pairs: possible targets (known objects of interest) at (within specified distance of?) the targeted location.
   i string: name or descriptor for known object
   ii lat-long

4 *Where, proportionately, is the {ech} level arty fire greatest?*
Answer returned: list of single, or multiple in the case of a tie (probably frequent, given the coarseness with which firing volume is specified), target taking the highest percentage of {ech} level arty fire, info for target(s) in format IIB. [*see assumptions]
a string: "Target of highest percentage of echelon-level arty fire"

b lat-long of location receiving fire.
c number: percentage: active fire on this location / potential able to hit this location
d number: percentage: active fire on this location / total active fire on board.
e string: highest echelon assets involved in firing on location. [*see assumptions]
f list of pairs: possible targets (known objects of interest) at (within specified distance of?) the targeted location.
  i string: name or descriptor for known object
  ii lat-long

C) <u>Changes in arty potential</u>

1 *Has any arty massing increased significantly (since the last time click)?*
Answer returned: list of massings that have increased more than 10% (if any), info for each massing in format IIC. [Increase is measured with respect to the percentage of the total arty that is concentrated in that spot, not the raw bomblets per minute numbers.]
a string: "Significant increase in massed arty potential"
b list: lat-long and radius (max range) for each unit involved
c number: percentage of increase since last time click
d number: percentage: total potential in this massing / total potential on board
e string: highest echelon assets involved in this massing {"Co"/"Bn"/"Bde"/"Div"}
f list of pairs: possible targets (known objects of interest) within the massed-on region. For each:
  i string: name or descriptor for known object
  ii lat-long

2 *Has {ech} arty massing on any location increased significantly more (since last time click)?*
Answer returned: list of {ech} level massings that have increased more than 10% (if any), info for each massing in format IIC. [Increase is measure with respect to the percentage of the total arty that is concentrated in that spot, not the raw bomblets per minute numbers.]
a string: "Significant increase in massed echelon-level arty potential"
b list: lat-long and radius (max range) for each unit involved
c number: percentage of increase since last time click
d number: percentage: total potential in this massing / total potential on board
e string: highest echelon assets involved in this massing {"Co"/"Bn"/"Bde"/"Div"}
f list of pairs: possible targets (known objects of interest) within the massed-on region. For each:
  i string: name or descriptor for known object

ii lat-long

D) Changes in arty fire

1 *Has arty fire on any target increased significantly (since last time click)?* [Note: given the coarseness with which volume of fire is reported, it may turn out that every reported increase is a significant one).

Answer returned: list of all arty targets on which fire has increased by more than 10% (if any), info for each target in format IID.

a string: "Significant increase in arty fire on target"
b lat-long of location receiving fire.
c number: percentage of increase since last time click
d number: percentage: active fire on this location / potential able to hit this location
e number: percentage: active fire on this location / total active fire on board.
f string: highest echelon assets involved in firing on location. [*see assumptions]
g list of pairs: possible targets (known objects of interest) at (within specified distance of?) the targeted location.
  i string: name or descriptor for known object
  ii lat-long

2 *Has {ech} arty fire on any target increased more than 10% (since last time click)?*

Answer returned: list of all {ech} level arty targets on which fire has increased by more than 10% (if any), info for each target in format IID. [*see assumptions]

a string: "Significant increase in echelon-level arty fire on target"
b lat-long of location receiving fire.
c number: percentage of increase since last time click
d number: percentage: active fire on this location / potential able to hit this location
e number: percentage: active fire on this location / total active fire on board.
f string: highest echelon assets involved in firing on location. [*see assumptions]
g list of pairs: possible targets (known objects of interest) at (within specified distance of?) the targeted location.
  i string: name or descriptor for known object
  ii lat-long

E) Other candidate arty snapshots, given a location specified at beginning to fill out CCIR template.

If we get a location, or set of locations, of interest at the outset, as well as a % level for the third question, and can enter these at the beginning of the run to specify these CCIR, we can run any of these additional queries (which utilize the

reasoning on which IA-D are based).*Generally, if the answer is negative (or more accurately, negative as far as Cyc can tell), the result will be an empty list.
CCIR Specifications: Locations of interest (values for {location}):

- *Clarksville Bridge*
- *Dover Bridge*
- *Natural Ford at middle*
- *Point at which Rt. 41A crosses border*
- *Value for {%}: 40*

1 *Is there any arty able to range {location}?*
Answer: list of arty units in range, with the following information for each:
a. string: "Arty in range of critical location"
b. list: lat-long and radius (max range) for each unit involved
c. number: percentage of arty potential on board that is in range of this location
d. string: highest echelon assets involved in this massing

2 *Is there any arty massing on {location}?*
Answer: (maximal) arty massing for which this location is a possible target, with the following info:
a string: "Massed arty potential on critical location"
b list: lat-long and radius (max range) for each unit involved
c number: percentage: total potential in this massing / total potential on board
d string: highest echelon assets involved in this massing

3 *Is more than {%} of the arty massed on {location}?*
Answer: (maximal) arty massing for which this location is a possible target, and which represents more than {%} of the arty on the board, with the following info:
a string: "Significant massing of arty potential on critical location"
b list: lat-long and radius (max range) for each unit involved
c number: percentage: total potential in this massing / total potential on board
d string: highest echelon assets involved in this massing

4 *Is there {ech} arty able to range {location}?*
Answer: list of {ech} level arty units able in range, with the following information for each:
a string: "{Ech} arty in range of critical location"
b list: lat-long and radius (max range) for each unit involved
c number: percentage of arty potential on board that is in range of this location
d string: highest echelon assets involved in this massing

5 *Is there {ech} arty massed on {location}?*

Answer: (maximal) {ech} level arty massing for which this location is a possible target, with the following info:

a string: "Massing of {ech} arty potential on critical location"
b list: lat-long and radius (max range) for each unit involved
c number: percentage: total potential in this massing / total potential on board
d string: highest echelon assets involved in this massing

6 *Is more than {%} of the {ech} arty massed on {location}?*

Answer: (maximal) {ech} level arty massing for which this location is a possible target, and which represents more than {%} of the {ech} level arty on the board, with the following info:

a string: "Significant massing of {ech} arty potential on critical location"
b list: lat-long and radius (max range) for each unit involved
c number: percentage: total potential in this massing / total potential on board
d string: highest echelon assets involved in this massing

7 *Is there arty fire on {location}?*

Answer: list of all current answers from IB1 for which {location} is (exactly? within some specified distance of?) the target location, with info for each in format IIB.

a string: "Arty firing on critical target"
b lat-long of location receiving fire.
c number: percentage: active fire on this location / potential able to hit this location
d number: percentage: active fire on this location / total active fire on board.
e string: highest echelon assets involved in firing on location. [*see assumptions]
f list of pairs: possible targets (known objects of interest) at (within specified distance of?) the targeted location.
  i string: name or descriptor for known object
  ii lat-long

8 *Is more than {%} of the arty in range of {location} actively firing?*

Answer: list of all answers from IB1 such that {location} is (exactly? within some specified distance of?) the target location, and the percentage of active/in-range arty is greater than {%}. Info for each in format IIB.

a string: "Significant percentage of in-range arty firing on critical target"
b lat-long of location receiving fire.
c number: percentage: active fire on this location / potential able to hit this location
d number: percentage: active fire on this location / total active fire on board.
e string: highest echelon assets involved in firing on location. [*see assumptions]

- f list of pairs: possible targets (known objects of interest) at (within specified distance of?) the targeted location.
  - i string: name or descriptor for known object
  - ii lat-long

9 *Is there {ech} arty fire on {location}?*

Answer: list of all current answers from IB3 for which {location} is (exactly? within some specified distance of?) the target location, with info for each in format IIB. [*see assumptions]

- a string: "Echelon-level arty firing on critical target"
- b lat-long of location receiving fire.
- c number: percentage: active fire on this location / potential able to hit this location
- d number: percentage: active fire on this location / total active fire on board.
- e string: highest echelon assets involved in firing on location. [*see assumptions]
- f list of pairs: possible targets (known objects of interest) at (within specified distance of?) the targeted location.
  - i string: name or descriptor for known object
  - ii lat-long

10 *Is more than {%} of the {ech} arty in range of {location} actively firing?*

Answer: list of all answers from IB3 such that {location} is (exactly? within some specified distance of?) the target location, and the percentage of active/in-range arty is greater than {%}. Info for each in format IIB. [*see assumptions]

- a string: "Significant percentage of in-range echelon-level arty firing on critical target"
- b lat-long of location receiving fire.
- c number: percentage: active fire on this location / potential able to hit this location
- d number: percentage: active fire on this location / total active fire on board.
- e string: highest echelon assets involved in firing on location. [*see assumptions]
- f list of pairs: possible targets (known objects of interest) at (within specified distance of?) the targeted location.
  - i string: name or descriptor for known object
  - ii lat-long

F) *Is any of these (IE, 1-10) true now that wasn't true in the last time-click?*

Answer: list of newly true answer objects. For each:

1 string: "New arty monitor condition met."
2 rest of answer formatted appropriately for query in question.

II) **Output formats**

A) Arty massing (intersection of arty ranges) info:
  1 string: description of what is significant about the massing
  2 list: lat-long and radius (max range) for each unit involved
  3 number: percentage: total potential in this massing / total potential on board
  4 string: highest echelon assets involved in this massing {"Co"/"Bn"/"Bde"/"Div"}
  5 list of pairs: possible targets (known objects of interest) within the massed-on region. For each:
    a string: name or descriptor for known object
    b lat-long

B) Target of arty fire info:
  1 string: description of what is significant about the target
  2 lat-long of location receiving fire.
  3 number: percentage: active fire on this location / potential able to hit this location
  4 number: percentage: active fire on this location / total active fire on board.
  5 string: highest echelon assets involved in firing on location. [*see assumptions]
  6 list of pairs: possible targets (known objects of interest) at (within specified distance of?) the targeted location.
    a string: name or descriptor for known object
    b lat-long

C) Arty massing change info:
  1 string: description of what is significant about the massing
  2 list: lat-long and radius (max range) for each unit involved
  3 number: percentage of increase since last time click
  4 number: percentage: total potential in this massing / total potential on board
  5 string: highest echelon assets involved in this massing {"Co"/"Bn"/"Bde"/"Div"}
  6 list of pairs: possible targets (known objects of interest) within the massed-on region. For each:
    a string: name or descriptor for known object
    b lat-long

D) Arty fire change info:
  1 string: description of what is significant about the target
  2 lat-long of location receiving fire.
  3 number: percentage of increase since last time click
  4 number: percentage: active fire on this location / potential able to hit this location
  5 number: percentage: active fire on this location / total active fire on board.
  6 string: highest echelon assets involved in firing on location. [*see assumptions]

7 list of pairs: possible targets (known objects of interest) at (within specified distance of?) the targeted location.
   a string: name or descriptor for known object
   b lat-long

III) **Underlying pieces: If desired, any of these can be included in the output to GITI, for use in further optional display.** ***Not desired; no interest in including them expressed.***

A) For each unit reported on in this time-click:
   1 Has it been identified and fused with any previously known unit? Answer: list of units (each such unit identified as subj/obj/source of a particular report)?
   2 What capabilities does it seem to have? Answer: list of strings, each representing a capability estimate)?

B) For each artillery unit reported on in this time-click:
   1 What echelon level asset does it appear to be? Answer: string, representing the highest echelon level asset detected.
   2 What is its bomblets per minute potential? Answer: number (total bpm of all arty pieces reported for unit).
   3 What % of the total arty potential on the board does it represent? Answer: number (total bpm for unit / total bpm on board for force).
   4 What is its max range? Answer: number (radius of max range, based on longest range weapon reported for unit).

Appendix C: Annotated Reconnaissance Monitors Output Description Document

[This is the tentative schedule drawn up during April meeting. Note that it depends on the timely completion of the Arty schedule above. As noted above in Appendix B, in fact the corrected and full battle data (i.e., including locations of reporting units) was not received by us until July, and that the arty tasks were therefore unable to get into full swing until after we had planned to be nearly done.]

Schedule

- **June 21, 2002**
  Components which are basically ready need only to set up queries and alerts
  Questions below: 1 (sans echelon), 6, 5 with some tweaking, 7, 8, 9
  Estimate 2 person/weeks, mostly OE effort
- **June 21, 2002**
  Add echelon reasoning
  Questions below: 2, enhanced version of 1
  Estimate 2 person/weeks, mostly coding effort
- **June 28, 2002**
  Add UAV LOS modifications
  Questions below: 3
  Estimate 2 person/weeks mostly coding effort, probably need Mike
- **June 28, 2002**
  Add Aggressiveness reasoning
  Questions below: 4
  Estimate 1 person/week mostly OE effort
- **Week of July 8th, 2002**
  BAT iteration
- **July 26th, 2002**
  Add current location filtering, specif. of objects of interest
  Questions below: 10, 11
  Estimate 3 person/days - 1 person/week effort
- **July 26th, 2002**
  Add recon metrics reasoning
  Questions below: 12, 13
  Estimate 2 person/days mostly OE effort
- **July 26th, 2002**
  Combine recon metrics and echelon reasoning
  Estimate 2 person/days mostly OE effort
- **July 26th, 2002**
  Version 2 of I-IV
- **Week of Aug 5, 2002**
  BAT iteration

- ***Remainder of August*, 2002**
  as needed for revisions and repairs.

I) <u>Questions to be answered</u>

1 *Where is the recon?*
Answer: list of recon units: For each unit:
a Location: lat-long and rad for los display
b Echelon
c List of known objects possibly seen
Break out blue units/ terrain
Noted during meeting: could possibly reason from attached special equipment, but there isn't any in this scenario.

2 *Where is the Bde-level recon?*
Answer: list of recon units with inferred echelon = Bde. For each unit:
a Location: lat-long and rad for los display
b Echelon
c List of known objects possibly seen
Break out blue units/ terrain
Noted during meeting: Not clear that we have any way of getting this out of the data that isn't just equivalent to (3).

3 *Where is the recon with special equipment type* X *(e.g., UAVs)?*
Answer: list of recon units with *X*: For each unit:
a Location: lat-long and rad for los display
b Echelon
c List of known objects possibly seen
Break out blue units/ terrain
Noted during meeting: UAVs may be only such equipment in scenario

4 *Where is recon most aggressive?*
Answer: list of recon units with highest aggressiveness rating. For each unit:
a Location: lat-long and rad for los display
b Echelon
c List of known objects possibly seen
Break out blue units/ terrain
Noted during meeting: Not clear that we have any good way of reasoning about aggressiveness from the data we've got, or that we can develop such reasoning in the time remaining.

5 *Where is the current (= within the past hour) recon focused?*
Answer: list of groupings of recon units possibly looking at the same area. For each grouping:

a list of tuples: {recon units / point- rad for los / list of known objects possibly seen by all / percentage of enemy recon assets focusing here now}

6 *Where have the recon assets been focused over time?*
Answer: list of groupings of recon units that have possibly looked at the same area. For each grouping:

a list of tuples: recon units / point- rad for los / time seen list of known objects possibly seen by all percentage of enemy recon assets focusing here over time

7 *Where are recon assets of multiple types currently/recently focused?*
Answer: list of multi-type groupings of recon units possibly looking at the same area For each grouping:

a list of tuples: recon units / type / point- rad for los list of known objects possibly seen by all percentage of enemy recon assets focusing here now

8 *Where have recon assets of multiple types been focused over time?*
Answer: list of multi-type groupings of recon units that have possibly looked at the same area For each grouping:

a list of tuples: recon units / type / point- rad for los / time seen list of known objects possibly seen by all percentage of enemy recon assets focusing here over time

9 *What known objects has the recon seen?*
Answer: list of objects possibly seen by some recon unit. For each object:

a list of encounter tuples: Recon unit / loc at time of sighting (lat-long, rad) / time seen

10 *Has* X *been seen in its current location (w/in footprint of unit)?*
X specified as CCIR at beginning of battle

a Answer: list of encounter tuples: Recon unit / loc at time of sighting (lat-long, rad) / time seen

11 *Is* X *currently seen?*
Answer: list of units possibly currently looking at *X* For each unit:

a Location: lat-long and rad for los display Echelon List of known objects possibly seen

Break out blue units/ terrain

12 *Where is the current/recent recon most focused?*
Answer: list of most significant grouping (or multiple groupings, if there's a tie) of recon units possibly looking at the same thing. For each grouping:

a list of tuples: recon units / point- rad for los list of known objects possibly seen by all percentage of enemy recon assets focusing here now

13 *Where has the recon been most concentrated over time?*

Answer: list of most significant grouping (or multiple groupings, if there's a tie) of recon units possibly looking at the same thing over time. For each grouping:

a list of tuples: recon units / point- rad for los list of known objects possibly seen by all percentage of enemy recon assets focusing here over time

14 *What percentage of Bde-level recon assets are currently focused on* X*?*

Answer:

a percentage of Bde-level recon currently focused on *X*

b list of tuples: Bde-recon units / point- rad for los list of known objects possibly seen by all

15 *What percentage of Bde-level recon assets have been focused on* X *over time?*

a Answer: percentage of Bde-level recon in LOS of *X* at any time

b list of tuples: Bde-recon units / point- rad for los / time of encounter list of known objects possibly seen by all

16 *Are more than 40% of the Bde recon assets currently focused on* X*?*

Answer: if measure of total Bde recon assets possibly seeing *X* is greater than 40% of the Bde recon assets we think that the enemy possesses, then give grouping of all such assets:

a list of tuples: recon units / point- rad for los list of known objects possibly seen by all

17 *Have more than 40% of the recon assets focused on* X *over time?*

Answer: if measure of total Bde recon assets possibly seeing *X* over time is greater than 40% of the Bde recon assets we think that the enemy possesses, then give grouping of all such assets:

a list of tuples: recon units / point- rad for los / time list of known objects possibly seen by all